x86 Registers

The main tools to write programs in x86 assembly are the processor registers. The registers are like variables built in the processor. Using registers instead of memory to store values makes the process faster and cleaner. The problem with the x86 serie of processors is that there are few registers to use. This section describes the main use of each register and ways to use them. That in note that the rules described here are more suggestions than strict rules. Some operations need absolutely some kind of registers but most of the you can use any of the freely.

Here is a list of the available registers on the 386 and higher processors. This list shows the 32 bit registers. Most of the can be broken down to 16 or even 8 bits register.

```
General registers
EAX EBX ECX EDX

Segment registers
CS DS ES FS GS SS

Index and pointers
ESI EDI EBP EIP ESP

Indicator
EFLAGS
```

General registers

As the title says, general register are the one we use most of the time Most of the instructions perform on these registers. They all can be broken down into 16 and 8 bit registers.

```
32 bits :  EAX EBX ECX EDX
16 bits : AX BX CX DX
 8 bits : AH AL BH BL CH CL DH DL
```

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

The "H" and "L" suffix on the 8 bit registers stand for high byte and low byte. With this out of the way, let's see their individual main use

```
EAX,AX,AH,AL : Called the Accumulator register.
               It is used for I/O port access, arithmetic, interrupt calls,
               etc...

EBX,BX,BH,BL : Called the Base register
               It is used as a base pointer for memory access
               Gets some interrupt return values

ECX,CX,CH,CL : Called the Counter register
               It is used as a loop counter and for shifts
               Gets some interrupt values

EDX,DX,DH,DL : Called the Data register
               It is used for I/O port access, arithmetic, some interrupt
               calls.
```

Segment registers

Segment registers hold the segment address of various items. They are only available in 16 values. They can only be set by a general register or special instructions. Some of them are critical for the good execution of the program and you might want to consider playing with them when you'll be ready for multi-segment programming

```
CS         : Holds the Code segment in which your program runs.
             Changing its value might make the computer hang.

DS         : Holds the Data segment that your program accesses.
             Changing its value might give erronous data.
```

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
ES,FS,GS   : These are extra segment registers available for
             far pointer addressing like video memory and such.

SS         : Holds the Stack segment your program uses.
             Sometimes has the same value as DS.
             Changing its value can give unpredictable results,
             mostly data related.
```

Indexes and pointers

Indexes and pointer and the offset part of and address. They have various uses but each register has a specific function. They some time used with a segment register to point to far address (in a 1Mb range). The register with an "E" prefix can only be used in protected mode.

```
ES:EDI EDI DI : Destination index register
                Used for string, memory array copying and setting and
                for far pointer addressing with ES

DS:ESI EDI SI : Source index register
                Used for string and memory array copying

SS:EBP EBP BP : Stack Base pointer register
                Holds the base address of the stack

SS:ESP ESP SP : Stack pointer register
                Holds the top address of the stack

CS:EIP EIP IP : Index Pointer
                Holds the offset of the next instruction
                It can only be read
```

The EFLAGS register

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

The EFLAGS register hold the state of the processor. It is modified by many intructions and is used for comparing some parameters, conditional loops and conditionnal jumps. Each bit holds the state of specific parameter of the last instruction. Here is a listing :

```
Bit   Label    Desciption
----------------------------
0      CF      Carry flag
2      PF      Parity flag
4      AF      Auxiliary carry flag
6      ZF      Zero flag
7      SF      Sign flag
8      TF      Trap flag
9      IF      Interrupt enable flag
10     DF      Direction flag
11     OF      Overflow flag
12-13  IOPL    I/O Priviledge level
14     NT      Nested task flag
16     RF      Resume flag
17     VM      Virtual 8086 mode flag
18     AC      Alignment check flag (486+)
19     VIF     Virutal interrupt flag
20     VIP     Virtual interrupt pending flag
21     ID      ID flag

Those that are not listed are reserved by Intel.
```

Undocumented registers

There are registers on the 80386 and higher processors that are not well documented by Intel. These are divided in control registers, debug registers, test registers and protected mode segmentation registers. As far as I know, the control registers, along with the segmentation registers, are used in protected mode programming, all of these registers are available on 80386 and higher processors except the test registers that have been removed on the pentium. Control registers are CR0 to CR4, Debug registers are DR0 to DR7, test registers are TR3 to TR7 and the

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

protected mode segmentation registers are GDTR (Global Descriptor Table Register), IDTR (Interrupt Descriptor Table Register), LDTR (Local DTR), and TR.

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com